

(京都学園大学経営学部論集 第19巻 第2号 2010年3月 1頁～11頁)

1

## 論 文

## OSS デスクトップの普及についての一考察

翁 長 朝 英

**要約** Linux 等の OSS デスクトップの普及の観点から、新しいパッケージ管理ツールやアプリケーションの利用方法について考察している。OSS デスクトップで採用されている従来のパッケージ管理ツールでは、ユーザーアプリケーションのインストールにも管理者権限を必要とし、またアプリケーションプログラムもシステムプログラムと同じ場所にインストールされる。ここでは、ユーザー権限でアプリケーションがインストールでき、簡単にアプリケーションを利用できるツールについて検討している。OSS デスクトップが一般ユーザーにも広く利用されるためには、OSS 利用に関する情報提供やサポート環境の充実とともに、このようなツールの普及も重要であることを指摘している。

**キーワード** OSS, デスクトップ, パッケージ管理

## はじめに

オープンソースソフトウェア (OSS) によるデスクトップ環境の普及を目指して、近年様々な取組みが行われている<sup>1)</sup>。日本 OSS 推進フォーラムでは、プラットフォーム部会にデスクトップ普及に関するタスクフォースを設置し、OSS デスクトップ利用の課題抽出や普及に向けた提言などの取組みを行っている<sup>2)</sup>。また、自治体における OpenOffice.org の導入や iPedia による OSS アプ

1) 飯尾淳「Linux デスクトップ活用の普及に向けた取り組み」Japan Linux Conference 2009 抄録集 Vol.3 日本 Linux 協会, 2009

2) 日本 OSS 推進フォーラム プラットフォーム部会 デスクトップ普及戦略検討タスクフォース「OSS デスクトップ普及促進に関する提言」  
<http://www.ipa.go.jp/software/open/forum/download/FY2008gambitTFoutput.pdf>, 2009

リケーションの情報提供などの取組みも行われている<sup>3) 4)</sup>。

このように、自治体などでは OSS デスクトップやアプリケーションの導入が進められているが、一般ユーザーも含めた OSS デスクトップの普及率は未だに低い状況である<sup>5) 6)</sup>。上述の提言にも記されているが、一般ユーザーによる OSS デスクトップ利用の普及には、幅広いレベルのユーザーが利用できる操作性とインタフェースの提供や多種多様なニーズに応えるアプリケーションのラインナップなどが重要である。

日本語環境も含めた操作性に関する技術的課題は、統合型デスクトップ環境やデスクトップ利用を目的としたディストリビューションの開発によって解決されつつある。例えば、Ubuntu ディストリビューションでは、「アプリケーションの追加と削除」ツールや Web を利用したアプリケーションのインストールなど、一般ユーザーでも扱い易いようなインストール方法の改善がなされている<sup>7) 8)</sup>。また、パッケージ管理システムのリポジトリには、様々なユーザーアプリケーションが揃えられている。

しかしながら、このようなディストリビューションで採用しているパッケージ管理システムは、ユーザーアプリケーションのインストールにも管理者権限が必要である。また、従来の Unix 型のファイル配置がとられ、ユーザーアプリケーションもシステム領域と同じ場所にインストールされる。これは、一般ユーザーには、インストールされたファイルの確認が困難なだけでなく、システムの安定性や安全性の面からも好ましいとは言えない。さらに、一つのプロ

---

3) 会津若松市「オープンオフィスと ODF 形式文書を導入しています」  
<http://www.city.aizuwakamatsu.fukushima.jp/ja/shisei/torikumi/ooo/index.htm>

4) OSS iPedia「オープンソース情報データベース」<http://ossipedia.ipa.go.jp/>

5) ThinkIT 監修「Linux オープンソース白書2006」インプレス、2005

6) Net Applications「Operating system market share」  
<http://marketshare.hitslink.com/>

7) Canonical <http://www.ubuntu.com/>

8) Appnr.com <http://appnr.com/>

グラムでは、異なるバージョンのプログラムをインストールすることができない問題もある。このように、一般ユーザーが種々のアプリケーションを手軽に利用できる状況にはなっていないように思われる。

一般ユーザーも対象とした OSS デスクトップの普及には、様々なアプリケーションを手軽に利用できるように、アプリケーションのインストールやアンインストールが容易にできることが重要である。また、GNUstep<sup>9)</sup>や ROX-Desktop<sup>10)</sup>で取り入れられているアプリケーションディレクトリのような、より分かり易いファイル配置も望まれる<sup>11)</sup>。さらに、システムの安定性や安全性を確保することも大切である。

現在、このような観点から、新しいパッケージ管理ツールの開発が行われている。Zero Install<sup>12)</sup>と Nix<sup>13)</sup>である。Nix はユーザーアプリケーションのみならず、NixOS ディストリビューションのパッケージ管理ツールとして、システムプログラムのインストールにも利用されている。また、パッケージ管理ツールの範疇には入らないが、一般ユーザーがアプリケーションを手軽に利用できるように開発された、Linux 用のポータブルアプリケーションがある。これには RUNZ<sup>14)</sup>技術が利用されている。

本稿では、ユーザーアプリケーションのインストールや利用に関して、これらのツールを技術的側面から比較検討し、OSS デスクトップ普及の観点から、一般ユーザーが手軽に利用し得るパッケージやアプリケーション管理ツールについて考察する。さらに、ユーザーアプリケーションのインストールや利用方

---

9) GNUstep.org <http://www.gnustep.org/>

10) ROX Desktop <http://roscidus.com/desktop/>

11) 翁長朝英「オープンソースソフトウェアによるデスクトップ環境の開発」  
『京都学園大学総合研究所』第9号, 2008

12) Zero Install <http://zero-install.sourceforge.net/>

13) NixOS.org <http://nixos.org/nix/>

14) Hacktolive.org <http://hacktolive.org/wiki/Hacktolive.org>

法の技術的な方向性について考察する。

## I ユーザーアプリケーションの管理と利用

### 1 Zero Install

Zero Install は、ROX-Desktop プロジェクトの一環として開発された技術である。このツールを利用するには、zeroinstall-injector パッケージが必要である。Debian や Ubuntu ではシステムのパッケージ管理ツールを用いてインストールできるが、リポジトリに入っているパッケージは古いバージョンのものである。最新版はダウンロードのページから利用可能である。また、他の Linux ディストリビューションや BSD 系 OS でも利用できる。このツールのインストールには管理者権限が必要である。

Zero Install 用のアプリケーションを利用するには、目的のアプリケーションのランチャ (launcher) を作成する。これは、ROX-Desktop と GNOME や KDE などの一般的なデスクトップでは操作に多少の違いがあるが、ネットワーク上の各アプリケーションのページから、フィード (feed) と呼ばれるアプリケーション情報が記述された URI (Uniform Resource Identifier) をアプリケーション追加のアイコンやウィンドウにドロップして行う。まず、アプリケーション情報がダウンロードされる。各プログラムには電子署名が施されているので、その認証が行われる。一度署名が認証されると、次回からは同じ署名の認証は行われない。続いて、アプリケーションと関連するライブラリがダウンロードされる。

ランチャが作成されると、一般のデスクトップでは自動的にアプリケーションメニューに登録されるので、そこからアプリケーションを起動する。一方、ROX-Desktop では、ランチャはアプリケーションディレクトリとしてユーザーのホームフォルダ内の Apps フォルダ等に保存するので、そこからアプリ

ケーションを起動する。これらのアプリケーションの利用には、管理者権限は必要としない。

ダウンロードされたプログラムは、ユーザーのホームフォルダ内の.cache/Oinstall.net フォルダに、アプリケーションディレクトリのように一つのフォルダにまとめられた状態で保存される。保存するフォルダ名には、ハッシュ値が使われ、各プログラム毎に固有の名前が付けられる。ハッシュ値の生成には主に、SHA-1 や SHA-1-NEW が使われており、16進数で40桁の名前になる。このハッシュ値によって、同一アプリケーションであっても、異なるバージョンを区別して利用することが可能となっている。

アプリケーションのアップデートや削除などの管理も、容易に行えるようになっている。一般のデスクトップでは、管理用ツールを用いてグラフィカルに行う。ROX-Desktop では、rox-filer から操作する。

## 2 Nix

Nix は、nix.org で開発されている純関数型のパッケージ管理システムである。このツールを用いてアプリケーションやライブラリなどのパッケージをインストールすると、各プログラムはアプリケーションディレクトリとして一つのフォルダにまとめられ、/nix/store フォルダに保存される。その際、フォルダ名には、パッケージ名やバージョン番号だけでなく、32桁のハッシュ値が付与された名前が付けられる。これにはパッケージ間の依存関係の情報も含まれている。これによって、一つのアプリケーションで複数バージョンの共存が可能となる。

また、パッケージのインストールやアップデートの際、従来のパッケージ管理ツールは、システムの状態を記録した設定ファイルを上書きする。しかしながら、Nix では、パッケージを記述している Nix エクスプレッションを既存の

設定ファイルに追加して、新しい設定ファイルを生成する。この仕組みによって、任意のシステムの状態にロールバックできる機能を実現している。

Nix パッケージ (nixpkgs) の管理ツールは Nix ホームページのダウンロードページから利用できる。最新版は nix-0.13 で、まだ開発版である。ソースのみならず様々な OS やディストリビューションのバイナリが揃えられている。これらのツールのインストールには、管理者権限が必要である。

アプリケーション等のパッケージのインストールは、コマンドラインから行う。Nix パッケージのチャンネル内に、バイナリがあればそれがインストールされ、なければソースからビルドしてインストールされる。パッケージのインストール、アップデートやアンインストールなどの操作によってシステムに変更が生じる度に、新しい user-environment フォルダが作成される。このフォルダには、これまでにインストールされた、/nix/store に保存されている各プログラムフォルダ内のファイルなどがリンクされている。このフォルダにもハッシュ値が付与されており、その時点でのシステムの状態を識別できるようになっている。

```

/nix/var/nix/profiles
  default —→ default-2-link
  default-1-link —→ user-environment
  default-2-link —→ user-environment

```

図 1 user-environment とプロファイル  
矢印はリンクを示す。

また、これらの user-environment フォルダは、図 1 に示してあるように、プロファイルとして /nix/var/nix/profiles フォルダ内にリンクされ、まとめられている。デフォルトのプロファイル (default) は、通常その時点での最新のプロファイルにリンクされている。プログラムの追加や削除などシステムに変更が生じた場合は、新しく生成された user-environment フォルダへのリン

ク（default-3-link）が作られる。そしてデフォルトプロファイルが、この新しい世代のプロファイルにリンクされる。各プロファイルは世代と呼ばれる。

デフォルトプロファイルは、最終的にユーザーのホームフォルダ内の.nix-profile フォルダにリンクされる。結果として、.nix-profile フォルダ内には、bin や lib などのフォルダが作成され、これまでにインストールされたプログラムの実行ファイルやライブラリなどがリンクされている。これらの実行ファイルやライブラリにアクセスできるように、PATH や LD\_LIBRARY\_PATH 環境変数を設定しなければならない。

この世代の概念によって、世代のロールバックや切替えが簡単に行えるようになっている。ロールバックには次のコマンドを使用する。

```
$ nix-env --rollback
```

これは、ユーザープロファイルのデフォルトを一代後退したプロファイルにリンクして行われる。任意の世代への切り替えは次のコマンドで行う。

```
$ nix-env -G [世代番号]
```

これは、指定された世代番号のプロファイルにデフォルトのプロファイルをリンクして行う。

アプリケーション等のパッケージの削除についても工夫が凝らされている。削除のコマンドが実行されると、そのプログラムが除かれた新しい user-environment とそのリンクであるプロファイルが作成される。しかしながら、パッケージ本体はそのまま残される。これは、ロールバックや他のユーザーのプロファイルからの参照の可能性を考慮している。ガベージ・コレクタ（garbage collector）のコマンドが実行され、このパッケージが他のプログラムから参照されていないことが確認されたとき、パッケージはシステムから完全に削除される。

### 3 RUNZ

アプリケーションを USB メモリなどの可搬媒体に入れ、そこから実行できるようにしたものが、ポータブルアプリケーションである。Linux 用のポータブルアプリケーションを実現する 1 つの技術として RUNZ がある。これはアプリケーションのインストーラとしても利用されるが、ここではポータブルアプリケーションの利用方法について検討する。

RUNZ アプリケーションは、runz の拡張子をもつ ISO フォーマットのイメージファイルになっている。これは、CD や DVD と同じフォーマットであり、圧縮はされていない。ファイルの内容にアクセスするには、マウントの操作が必要となるが、Unix 系 OS では、通常ファイルシステムのマウントは管理者権限で行うようになっている。RUNZ では、ユーザーでもマウントできるように、ユーザー空間でファイルシステムを扱うことが可能な fuse を利用している。

RUNZ アプリケーションの実行には、RUNZ フレームワークが必要である。これはシステムツールとしてインストールするので、管理者権限が必要である。RUNZ アプリケーションには、GUI ツールキットとして GTK + や Qt が使用されている。そのため、このアプリケーションを利用するには、あらかじめこれらのツールキットもインストールする必要がある。

具体的なアプリケーションの実行方法は、以下の通りである。

- (1) ファイルブラウザから、目的の runz ファイルをダブルクリックする。
- (2) runz ファイルが /tmp/runz\_USERNAME/mounted フォルダにマウントされる。
- (3) マウントされたフォルダ内の autorun スクリプトが実行される。このスクリプト内には、アプリケーションの起動コマンドが記述されており、目的のアプリケーションが起動する。



(4) アプリケーションを終了すると自動的にアンマウントされる。

runz ファイルがマウントされる前に、セキュリティに関するダイアログが表示される。これら一連の処理は、RUNZ フレームワークの runz と runz-launch シェルスクリプトによって行われている。

ライブラリへのアクセスが必要な場合は、LD\_LIBRARY\_PATH 環境変数にライブラリの場所を設定しなければならない。これは runz ファイル内のアプリケーション起動スクリプト内で行われており、ユーザーが設定する必要はない。

ポータブルアプリケーションでは、アプリケーションの設定ファイルもアプリケーションと同じ場所に置かれることが望まれる。RUNZ では、アプリケーションと同じ場所に runz フォルダがあれば、そこに設定ファイルを格納するようになっている。しかしながら、設定ファイルをユーザーホームフォルダに格納するアプリケーションもある。

## Ⅱ 考 察

ここでは、前章で取り上げた3つのツールの特徴について考察する。

従来のパッケージ管理システムにはない、Zero Install と Nix に共通の特徴として、

- ・ユーザー権限でインストールが可能
- ・同一アプリケーションで複数バージョンが利用可能

な点が挙げられる。また、Zero Install の特徴として、アプリケーションのインストールやアップデート等すべての操作が GUI で行えることが挙げられる。特に、アプリケーションのインストールが、ネットワーク上にある URI をウインドウやアイコンにドロップするだけで行えるのは、Zero Install の優れた特徴である。

Nix の他のツールにない優れた特徴として、世代の概念がある。これによって、世代のロールバックや任意の世代への切替えが容易に行え、システムの状態を簡単に変更することができる。また、アプリケーションの削除の方法にも工夫がなされ、より安全なアプリケーション管理が可能である。しかしながら、現バージョンにおける Nix の操作はコマンドラインから行うようになっており、GUI 操作が強く望まれる。

また、本稿ではアプリケーションの手軽な利用という観点から、Linux 用のポータブルアプリケーションを実現するための RUNZ 技術についても検討した。RUNZ アプリケーションの利用における大きな特徴は、インストールの操作がいらない点である。任意のフォルダに runz ファイルを置き、これをダブルクリックするだけでアプリケーションを起動し利用することができる。この技術は近年開発が始められたところであり、まだ利用可能なアプリケーションの数も少ないが、利用の手軽さから今後普及するであろうと期待される。

RUNZ と類似した技術に <sup>15)</sup>Splashtop がある。これは、ネットブック PC に主に採用されている技術である。Splashtop では、アプリケーションを squashfs ファイルシステムとして 1 つのイメージファイルにまとめ、これをシステムの起動プロセスの中でマウントして利用する。squashfs のイメージファイルのマウントには管理者権限が必要である。

このように、ファイルシステムを用いた新たなアプリケーションの利用方法は、今後さらに普及するであろうと思われる。しかしながら、これには最近開発された技術が取り入れられており、問題も見受けられる。特に、squashfs の場合は、ファイルシステム作成ツールのバージョン間に互換性がなく、ファイルシステムのマウント時にエラーが起き易いという問題がある。

---

15) DeviceVM <http://www.splashtop.com/index.php>

## おわりに

本稿では OSS デスクトップ利用の普及の観点から、ユーザーアプリケーションを手軽にインストールし利用するためのツールについて考察した。

これまでに多くのツールが考案されているが、このようなツールには新しい技術が取り入れられており、これらの技術の普及も OSS デスクトップ普及の大きな要因となっている。

また、本稿で検討したツールは、それぞれに固有な特徴を有している。これらのツールが、今後一般ユーザーにも広く利用され普及するためには、より簡単にアプリケーションを実行できることが重要であろう。

### 《参考文献》

1. 翁長朝英「GNUstep 入門」ふくろう出版, 2007
  2. 翁長朝英「オープンソースソフトウェアによるデスクトップ環境の開発」『京都学園大学総合研究所』第 9 号, 2008
  3. E. Dolstra, A. Lon, NixOS: A Purely Functional Linux Distribution, ICFP2008, Victoria, 2008
  4. E. Dolstra, M. Jonge, E. Visser, Nix: A Safe and Policy-Free System for Software Deployment, LISA XVIII Atlanta, 2004
  5. MOONGIFT「マルチプラットフォームで動作するパッケージ管理システム『Zero Install』」[http://www.moongift.jp/2008/08/zero\\_install/](http://www.moongift.jp/2008/08/zero_install/)
  6. 西村賢@IT「[[Top10] さらば DLL 地獄, 純関数型 OS『NixOS』」  
<http://www.atmarkit.co.jp/news/200902/17/nixos.html>
  7. Zero Install <http://zero-install.sourceforge.net/doc.html> 各種ドキュメント
  8. NixOS.org <http://nixos.org/nixos/docs.html> 各種ドキュメント
  9. Hacktolive.org <http://hacktolive.org/wiki/RUNZ> 各種ドキュメント
- ・本論文に出てくる製品名等は、各社の商標および登録商標です。